

Міністерство освіти і науки України  
Національний технічний університет України «КПІ» імені Ігоря Сікорського  
Кафедра обчислювальної техніки ФІОТ

ЗВІТ  
з лабораторної роботи №5  
з навчальної дисципліни «Методи наукових досліджень»

Тема:

ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ  
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ  
КВАДРАТИЧНИХ ЧЛЕНІВ (ЦЕНТРАЛЬНИЙ ОРТОГОНАЛЬНИЙ  
КОМПОЗИЦІЙНИЙ ПЛАН)

Виконав:  
Студент групи ІВ-92,  
Карпека Дмитрій Юрійович

Перевірив:  
Регіда П. Г.

Київ 2021

**Мета роботи:** Провести трьохфакторний експеримент з урахуванням квадратичних членів ,використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису об'єкту.

### Завдання

1. Взяти рівняння з урахуванням квадратичних членів.
2. Скласти матрицю планування для ОЦКП
3. Провести експеримент у всіх точках факторного простору (знайти значення функції відгуку Y). Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі. Варіанти вибираються по номеру в списку в журналі викладача.

$$y_{i\max} = 200 + x_{i\max}$$

$$y_{i\min} = 200 + x_{i\min}$$

$$\text{где } x_{i\max} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{i\min} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

Варіант:

210	-6	10	-10	5	-9	3
-----	----	----	-----	---	----	---

Виконання роботи:

#### 1) Результати роботи програми:

Матриця повного факторного експерименту з натуралізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	X1**2	X2**2	X3**2	Y1	Y2	Y3	Aver Y	S_y
1	-6	-10	-9	60	54	90	-540	36	100	81	198	204	194	198.67	16.89
1	-6	-10	3	60	-18	-30	180	36	100	9	203	199	206	202.67	8.22
1	-6	5	-9	-30	54	-45	270	36	25	81	196	200	201	199.0	4.67
1	-6	5	3	-30	-18	15	-90	36	25	9	193	204	194	197.0	24.67
1	10	-10	-9	-100	-90	90	900	100	100	81	202	201	193	198.67	16.22
1	10	-10	3	-100	30	-30	-300	100	100	9	202	205	203	203.33	1.56
1	10	5	-9	50	-90	-45	-450	100	25	81	206	201	204	203.67	4.22
1	10	5	3	50	30	15	150	100	25	9	203	194	194	197.0	18.0
1	-7.72	-2.5	-3.0	19.3	23.16	7.5	-57.9	59.598	6.25	9.0	194	204	202	200.0	18.67
1	11.72	-2.5	-3.0	-29.3	-35.16	7.5	87.9	137.358	6.25	9.0	195	204	193	197.33	22.89
1	2.0	-11.613	-3.0	-23.23	-6.0	34.84	69.68	4.0	134.862	9.0	204	201	202	202.33	1.56
1	2.0	6.613	-3.0	13.23	-6.0	-19.84	-39.68	4.0	43.732	9.0	203	204	201	202.67	1.56
1	2.0	-2.5	-10.29	-5.0	-20.58	25.72	51.45	4.0	6.25	105.884	200	198	205	201.0	8.67
1	2.0	-2.5	4.29	-5.0	8.58	-10.72	-21.45	4.0	6.25	18.404	199	199	206	201.33	10.89
1	2.0	-2.5	-3.0	-5.0	-6.0	7.5	15.0	4.0	6.25	9.0	193	199	193	195.0	8.0

Матриця повного факторного експерименту з нормалізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	X1**2	X2**2	X3**2	Y1	Y2	Y3	Aver Y	S_y
1	-1	-1	-1	1	1	1	-1	1	1	1	198	204	194	198.67	16.89
1	-1	-1	1	1	-1	-1	1	1	1	1	203	199	206	202.67	8.22
1	-1	1	-1	-1	1	-1	1	1	1	1	196	200	201	199.0	4.67
1	-1	1	1	-1	-1	1	-1	1	1	1	193	204	194	197.0	24.67
1	1	-1	-1	-1	-1	1	1	1	1	1	202	201	193	198.67	16.22
1	1	-1	1	1	-1	1	-1	1	1	1	202	205	203	203.33	1.56
1	1	1	-1	1	-1	-1	-1	1	1	1	206	201	204	203.67	4.22
1	1	1	1	1	1	1	1	1	1	1	203	194	194	197.0	18.0
1	-1.215	0	0	-0.0	-0.0	0	-0.0	1.476	0	0	194	204	202	200.0	18.67

1	-1.215	0	0	-0.0	-0.0	0	-0.0	1.476	0	0	194	204	202	200.0	18.67
1	1.215	0	0	0.0	0.0	0	0.0	1.476	0	0	195	204	193	197.33	22.89
1	0	-1.215	0	-0.0	0	-0.0	-0.0	0	1.476	0	204	201	202	202.33	1.56
1	0	1.215	0	0.0	0	0.0	0.0	0	1.476	0	203	204	201	202.67	1.56
1	0	0	-1.215	0	-0.0	-0.0	-0.0	0	0	1.476	200	198	205	201.0	8.67
1	0	0	1.215	0	0.0	0.0	0.0	0	0	1.476	199	199	206	201.33	10.89
1	0	0	0	0	0	0	0	0	0	0	193	199	193	195.0	8.0

Рівняння

$y = 198.7 + 0.19 * x_1 + -0.57 * x_2 + 0.04 * x_3 + 0.5 * x_{1x2} + -0.5 * x_{1x3} + -2.17 * x_{2x3} + -0.67 * x_{1x2x3} + -0.85 * x_{1**2} + 1.75 * x_{2**2} + 0.85 * x_{3**2}$

Критерій Кохрена

Тест Кохрена: Gr = 0.148

Дисперсії однорідні з імовірністю 0.95

Критерій Ст'юдента

Коефіцієнт b1 незначимий, тому виключається із рівняння регресії

Коефіцієнт b2 незначимий, тому виключається із рівняння регресії

Коефіцієнт b3 незначимий, тому виключається із рівняння регресії

Коефіцієнт b4 незначимий, тому виключається із рівняння регресії

Коефіцієнт b5 незначимий, тому виключається із рівняння регресії

Коефіцієнт b6 незначимий, тому виключається із рівняння регресії

Коефіцієнт b7 незначимий, тому виключається із рівняння регресії

Скореговане рівняння регресії:

$y = 198.7 + 0.0 * x_1 + 0.0 * x_2 + 0.0 * x_3 + 0.0 * x_{1x2} + 0.0 * x_{1x3} + 0.0 * x_{2x3} + 0.0 * x_{1x2x3} + -0.85 * x_{1**2} + 1.75 * x_{2**2} + 0.85 * x_{3**2}$

Критерій Фішера

Математична модель неадекватна експериментальним даним на рівні значимості 0.05.

Повтор експерименту для m+1

Матриця повного факторного експерименту з натуралізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	X1**2	X2**2	X3**2	Y1	Y2	Y3	Y4	Aver Y	S_y
1	-6	-10	-9	60	54	90	-540	36	100	81	200	202	195	199	199.0	6.5
1	-6	-10	3	60	-18	-30	180	36	100	9	200	192	205	198	198.75	21.69
1	-6	5	-9	-30	54	-45	270	36	25	81	205	192	204	198	199.75	27.19
1	-6	5	3	-30	-18	15	-90	36	25	9	194	199	205	206	201.0	23.5
1	10	-10	-9	-100	-90	90	900	100	100	81	205	192	206	203	201.5	31.25
1	10	-10	3	-100	30	-30	-300	100	100	9	201	196	204	193	198.5	18.25
1	10	5	-9	50	-90	-45	-450	100	25	81	194	199	206	199	199.5	18.25
1	10	5	3	50	30	15	150	100	25	9	205	202	202	193	200.5	20.25
1	-7.72	-2.5	-3.0	19.3	23.16	7.5	-57.9	59.598	6.25	9.0	197	192	206	192	196.75	32.69
1	11.72	-2.5	-3.0	-29.3	-35.16	7.5	87.9	137.358	6.25	9.0	198	197	193	194	195.5	4.25
1	2.0	-11.613	-3.0	-23.23	-6.0	34.84	69.68	4.0	134.862	9.0	195	198	200	201	198.5	5.25
1	2.0	6.613	-3.0	13.23	-6.0	-19.84	-39.68	4.0	43.732	9.0	205	206	201	198	202.5	10.25
1	2.0	-2.5	-10.29	-5.0	-20.58	25.72	51.45	4.0	6.25	105.884	196	204	199	204	200.75	11.69
1	2.0	-2.5	4.29	-5.0	8.58	-10.72	-21.45	4.0	6.25	18.404	204	200	194	197	198.75	13.69
1	2.0	-2.5	-3.0	-5.0	-6.0	7.5	15.0	4.0	6.25	9.0	198	201	203	198	200.0	4.5

Матриця повного факторного експерименту з нормалізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	X1**2	X2**2	X3**2	Y1	Y2	Y3	Y4	Aver Y	S_y
1	-1	-1	-1	1	1	1	-1	1	1	1	200	202	195	199	199.0	6.5
1	-1	-1	1	1	-1	-1	1	1	1	1	200	192	205	198	198.75	21.69
1	-1	1	-1	-1	1	-1	1	1	1	1	205	192	204	198	199.75	27.19
1	-1	1	1	1	-1	1	-1	1	1	1	194	199	205	206	201.0	23.5
1	1	-1	-1	-1	-1	1	1	1	1	1	205	192	206	203	201.5	31.25
1	1	-1	1	-1	1	-1	-1	1	1	1	201	196	204	193	198.5	18.25
1	1	1	-1	1	1	-1	-1	1	1	1	194	199	206	199	199.5	18.25
1	1	1	1	1	1	1	1	1	1	1	205	202	202	193	200.5	20.25
1	-1.215	0	0	-0.0	-0.0	0	-0.0	1.476	0	0	197	192	206	192	196.75	32.69
1	1.215	0	0	0.0	0.0	0	0.0	1.476	0	0	198	197	193	194	195.5	4.25
1	0	-1.215	0	-0.0	0	-0.0	-0.0	0	1.476	0	195	198	200	201	198.5	5.25
1	0	1.215	0	0.0	0	0.0	0.0	0	1.476	0	205	206	201	198	202.5	10.25

1	1.215	0	0	0.0	0.0	0	0.0	0.0	1.476	0	0	198	197	193	194	195.5	4.25
1	0	-1.215	0	-0.0	0	-0.0	-0.0	0	1.476	0	0	195	198	200	201	198.5	5.25
1	0	1.215	0	0.0	0	0.0	0.0	0	1.476	0	0	205	206	201	198	202.5	10.25
1	0	0	-1.215	0	-0.0	-0.0	-0.0	0	0	1.476	196	204	199	204	200.75	11.69	
1	0	0	1.215	0	0.0	0.0	0.0	0	0	1.476	204	200	194	197	198.75	13.69	
1	0	0	0	0	0	0	0	0	0	0	198	201	203	198	200.0	4.5	

Рівняння

y = 198.75 + -0.0 \* x1 + 0.72 \* x2 + -0.31 \* x3 + -0.37 \* x1x2 + -0.37 \* x1x3 + 0.69 \* x2x3 + 0.31 \* x1x2x3 + -1.5 \* x1\*\*2 + 1.46 \* x2\*\*2 + 0.95 \* x3\*\*2

Критерій Кохрена

Тест Кохрена: G<sub>r</sub> = 0.131

Дисперсії однорідні з імовірністю 0.95

Критерій Ст'юдента

Коефіцієнт b1 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b2 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b3 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b4 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b5 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b6 незначимий, тому виключається із рівняння регресії  
Коефіцієнт b7 незначимий, тому виключається із рівняння регресії

Скореговане рівняння регресії:

y = 198.75 + 0.0 \* x1 + 0.0 \* x2 + 0.0 \* x3 + 0.0 \* x1x2 + 0.0 \* x1x3 + 0.0 \* x2x3 + 0.0 \* x1x2x3 + -1.5 \* x1\*\*2 + 1.46 \* x2\*\*2 + 0.95 \* x3\*\*2

Критерій Фішера

Математична модель адекватна експериментальним даним на рівні значущості 0.05

Process finished with exit code 0

### Код програми:

```

from random import randint
import prettytable
import math
from scipy.stats import f, t
from sklearn import linear_model
from functools import partial

# Лабораторна робота №5 "ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ
ВИКОРИСТАННІ
# РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ КВАДРАТИЧНИХ ЧЛЕНІВ (ЦЕНТРАЛЬНИЙ ОРТОГОНАЛЬНИЙ
КОМПОЗИЦІЙНИЙ ПЛАН)" з предмету МОПЕ
# Варіант №210 Карпека Дмитрій

variant = 210

min_x = [-6, -10, -9]
max_x = [10, 5, 3]

x0 = [1]
norm_x=[ [-1, -1, -1],
          [-1, -1, 1],
          [-1, 1, -1],
          [-1, 1, 1],
          [1, -1, -1],
          [1, -1, 1],
          [1, 1, -1],
          [1, 1, 1],
          [-1.215, 0, 0],
          [1.215, 0, 0],
          [0, -1.215, 0],
          [0, 1.215, 0],
          [0, 0, -1.215],

```

```
[0, 0, 1.215],
[0, 0, 0]]
```

```
x0i = list(map(lambda xmax, xmin: (xmax + xmin)/2, max_x, min_x))
delta_xi = [xmax - xi for xmax, xi in zip(max_x, x0i)]
```

```
natur_x = [ [min_x[0], min_x[1], min_x[2]],
             [min_x[0], min_x[1], max_x[2]],
             [min_x[0], max_x[1], min_x[2]],
             [min_x[0], max_x[1], max_x[2]],
             [max_x[0], min_x[1], min_x[2]],
             [max_x[0], min_x[1], max_x[2]],
             [max_x[0], max_x[1], min_x[2]],
             [max_x[0], max_x[1], max_x[2]],
             [round(-1.215*delta_xi[0]+x0i[0], 3), x0i[1], x0i[2]],
             [round(1.215*delta_xi[0]+x0i[0], 3), x0i[1], x0i[2]],
             [x0i[0], round(-1.215*delta_xi[1]+x0i[1], 3), x0i[2]],
             [x0i[0], round(1.215*delta_xi[1]+x0i[1], 3), x0i[2]],
             [x0i[0], x0i[1], round(-1.215*delta_xi[2]+x0i[2], 3)],
             [x0i[0], x0i[1], round(1.215*delta_xi[2]+x0i[2], 3)],
             [x0i[0], x0i[1], x0i[2]]]
```

```
def experiment(m=3, n=15):
```

```
    regression_str = 'y = {} + {} * x1 + {} * x2 + {} * x3 + {} * x1x2 + {} *
x1x3 + {} * x2x3 + {} * x1x2x3 + {} * x1**2 + {} * x2**2 + {} * x3**2'
```

```
    def matrix_plan(m, ymin, ymax, n):
        return [[randint(ymin, ymax) for _ in range(m)] for _ in range(n)]
```

```
    def multiplication(a, b):
        return a * b
```

```
    def average(list):
        return sum(list) / len(list)
```

```
    def round_to_2(number):
        return round(number, 2)
```

```
    def dispersion(list_y, aver_list_y):
        return [round_to_2(average(list(map(lambda y: (y - aver_list_y[i]) ** 2,
list_y[i])))) for i in range(len(list_y))]
```

```
    def cochrane_criteria(S_y):
        global m
        print("\nКритерій Кохрена\n")
        Gp = max(S_y) / sum(S_y)
        q = 0.05
        q_ = q / f2
        chr = f.ppf(q=1 - q_, dfn=f1, dfd=(f2 - 1) * f1)
        Gt = chr / (chr + f2 - 1)
        print("Тест Кохрена: Gr = " + str(round(Gp, 3)))
        if Gp < Gt:
            print("Дисперсії однорідні з імовірністю 0.95")
            pass
        else:
            print("\nДисперсії неоднорідні.\nПовтор експерименту для m + 1\n")
            m = m + 1
```

```

        experiment(m)

def student_criteria(S_y, d):
    print("\nКритерій Ст'юдента\n")
    bettaList = [sum(S_y) * x0[0] / n,
                  average(list(map(multiplication, S_y, x1i))),
                  average(list(map(multiplication, S_y, x2i))),
                  average(list(map(multiplication, S_y, x3i))),
                  average(list(map(multiplication, S_y, norm_x12))),
                  average(list(map(multiplication, S_y, norm_x13))),
                  average(list(map(multiplication, S_y, norm_x23))),
                  average(list(map(multiplication, S_y, norm_x123))),
                  average(list(map(multiplication, S_y, norm_sq_x1))),
                  average(list(map(multiplication, S_y, norm_sq_x2))),
                  average(list(map(multiplication, S_y, norm_sq_x3)))]
    bettaList = [round_to_2(i) for i in bettaList]

    list_t = [bettaList[i] * S for i in range(n-4)]

    for i in range(n-4):
        if list_t[i] < t.ppf(q=0.975, df=f3):
            list_b[i] = 0
            d -= 1
            print('Коефіцієнт b' + str(i) + ' незначимий, тому виключається із рівняння регресії')
        print("\nСкореговане рівняння регресії:")
        print(regression_str.format(*map(round_to_2, list_b)))

def fisher_criteria(d):
    global m
    print("\nКритерій Фішера\n")
    f4 = n - d
    S_ad = (m * sum(
        [(list_b[0] + list_b[1] * x1i[i] + list_b[2] * x2i[i] + list_b[3] *
        x3i[i] + list_b[4] * norm_x12[i] +
        list_b[5] * norm_x13[i] + list_b[6] * norm_x23[i] + list_b[7] *
        norm_x123[i] + list_b[8] * norm_sq_x1[i] + list_b[9] * norm_sq_x2[i] +
        list_b[10] * norm_sq_x3[i]
        - average_list_y[i]) ** 2 for i in range(n)]) / f4)
    Fp = S_ad / Sb

    if Fp > f.ppf(q=0.95, dfn=f4, dfd=f3): # перевірка критерію Фішера з використанням scipy
        print('Математична модель неадекватна експериментальним даним на рівні значимості 0.05.\nПовтор експерименту для m+1')
        m = m + 1
        experiment(m)
    else:
        print('Математична модель адекватна експериментальним даним на рівні значущості 0.05')

def printed_matrixes():
    pt1 = prettytable.PrettyTable()
    pt2 = prettytable.PrettyTable()
    pt1.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23", "X123"] + ["X1**2", "X2**2", "X3**2"] + ["Y" + str(x) for x in range(1, m + 1)] + ["Aver Y"] + ["S_y"]
    pt2.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23", "X123"] + ["X1**2", "X2**2", "X3**2"] + ["Y" + str(x) for x in range(1, m + 1)]

```

```

+ ["Aver Y"] + ["S_y"]

    print("Матриця повного факторного експерименту з натуралізованими
значеннями:\n")
    pt1.add_rows([x0 + natur_x[i] + natur_x12[i] + natur_x13[i] +
natur_x23[i] + natur_x123[i] + [natur_sq_x1[i], natur_sq_x2[i], natur_sq_x3[i]]
+ matrix_y[i] + [average_list_y[i]] + [S_y[i]] for i in range(n)])
    print(pt1)

    print("\nМатриця повного факторного експерименту з нормалізованими
значеннями:\n")
    pt2.add_rows([x0 + norm_x[i] + [norm_x12[i]] + [norm_x13[i]] +
[norm_x23[i]] + [norm_x123[i]] + [norm_sq_x1[i], norm_sq_x2[i], norm_sq_x3[i]] +
matrix_y[i] + [average_list_y[i]] + [S_y[i]] for i in range(n)])
    print(pt2)

m = m
n = 15

x_average_max = sum(max_x) / 3
x_average_min = sum(min_x) / 3

y_max = round(200 + x_average_max)
y_min = round(200 + x_average_min)

matrix_y = matrix_plan(m, y_min, y_max, n)
average_list_y = [round(average(matrix_y[i]), 2) for i in
range(len(matrix_y))]

S_y = dispersion(matrix_y, average_list_y)

f1 = m - 1
f2 = n
f3 = f1 * f2
d = 11

Sb = sum(S_y) / n
S = math.sqrt(Sb / (n * m))

norm_x12 = [round(norm_x[i][0] * norm_x[i][1], 3) for i in
range(len(norm_x))]
norm_x13 = [round(norm_x[i][0] * norm_x[i][2], 3) for i in
range(len(norm_x))]
norm_x23 = [round(norm_x[i][1] * norm_x[i][2], 3) for i in
range(len(norm_x))]
norm_x123 = [round(norm_x[i][0] * norm_x[i][1] * norm_x[i][2], 3) for i in
range(len(norm_x))]
#--
norm_sq_x1 = [round(norm_x[i][0]**2, 3) for i in range(len(norm_x))]
norm_sq_x2 = [round(norm_x[i][1]**2, 3) for i in range(len(norm_x))]
norm_sq_x3 = [round(norm_x[i][2]**2, 3) for i in range(len(norm_x))]
#--

natur_x12 = [[round_to_2(natur_x[i][0] * natur_x[i][1])] for i in
range(len(natur_x))]
natur_x13 = [[round_to_2(natur_x[i][0] * natur_x[i][2])] for i in
range(len(natur_x))]
natur_x23 = [[round_to_2(natur_x[i][1] * natur_x[i][2])] for i in
range(len(natur_x))]

```

```

    natur_x123 = [[round_to_2(natur_x[i][0] * natur_x[i][1] * natur_x[i][2])]
for i in range(len(natur_x))]

    natur_sq_x1 = [round(natur_x[i][0]**2, 3) for i in range(len(natur_x))]
    natur_sq_x2 = [round(natur_x[i][1]**2, 3) for i in range(len(natur_x))]
    natur_sq_x3 = [round(natur_x[i][2]**2, 3) for i in range(len(natur_x))]

    x1i = [norm_x[i][0] for i in range(n)]
    x2i = [norm_x[i][1] for i in range(n)]
    x3i = [norm_x[i][2] for i in range(n)]

    # b0, b1, b2, b3, b12, b13, b23, b123, b11, b22, b33
    list_for_b = [x0 + norm_x[i] + [norm_x12[i]] + [norm_x13[i]] + [norm_x23[i]]
+ [norm_x123[i]] + [norm_sq_x1[i], norm_sq_x2[i], norm_sq_x3[i]] for i in
range(n)]

    skm = linear_model.LinearRegression(fit_intercept=False)
    skm.fit(list_for_b, average_list_y)
    list_b = skm.coef_

    printed_matrixes()
    print("\nPівняння\n" + regression_str.format(*map(round_to_2, list_b)))
    cochrane_criteria(S_y)
    student_criteria(S_y, d)
    fisher_criteria(d)
#-----
m = 3
experiment(m)

```